Desarrollo de SwiftGadget: Una plataforma Web para la Venta de Dispositivos Tecnológicos.

Juan David Bejarano Díaz

Escuela Tecnológica Instituto Técnico Central

Facultad de Sistemas

Electiva Técnica PHP

Ing. Linda Françoise Caicedo Morea

Bogotá D.C, Colombia

4 de marzo de 2025

Contenido

[Planteamiento Del Problema 4](#_Toc191986659)

[Pregunta De Investigación 6](#_Toc191986660)

[Objetivos 7](#_Toc191986661)

[Objetivo general 7](#_Toc191986662)

[Objetivos específicos 7](#_Toc191986663)

[Introducción 8](#_Toc191986664)

[Análisis de Requerimientos 11](#_Toc191986665)

[Objetivo del Proyecto 11](#_Toc191986666)

[Requisitos Funcionales 11](#_Toc191986667)

[Requisitos no Funcionales 12](#_Toc191986668)

[Diseño 12](#_Toc191986669)

[Arquitectura del Sistema 12](#_Toc191986670)

[Cliente-Servidor 13](#_Toc191986671)

[Diseño de Interfaz de Usuario (UI/UX) 13](#_Toc191986672)

[Experiencia de Usuario (UX) 14](#_Toc191986673)

[Herramientas y Tecnologías 14](#_Toc191986674)

[Estrategia de Desarrollo 14](#_Toc191986675)

[Seguridad en el Diseño 15](#_Toc191986676)

[Proceso de Desarrollo 16](#_Toc191986677)

[Planificación: 16](#_Toc191986678)

[Desarrollo Ágil: 16](#_Toc191986679)

[Pruebas: 16](#_Toc191986680)

[Documentación: 16](#_Toc191986681)

[Revisar y Adaptar: 16](#_Toc191986682)

[Prueba 17](#_Toc191986683)

[Tipo de Prueba 17](#_Toc191986684)

[Implementación 17](#_Toc191986685)

[Preparación de la Implementación: 17](#_Toc191986686)

[Estrategia de Lanzamiento: 17](#_Toc191986687)

[Monitoreo Posterior a la Implementación: 17](#_Toc191986688)

[Mantenimiento 17](#_Toc191986689)

[Programa de Mantenimiento: 17](#_Toc191986690)

[Documentación continua: 18](#_Toc191986691)

[Atención al cliente: 18](#_Toc191986692)

[Referencias 19](#_Toc191986693)

# Planteamiento Del Problema

El desarrollo de plataformas de comercio electrónico ha crecido significativamente en los últimos años, permitiendo a muchas empresas ofrecer productos y servicios de manera eficiente. Sin embargo, este crecimiento también ha traído consigo desafíos que afectan la funcionalidad y el éxito de estos sitios web.

Por ende, el desarrollo de una plataforma web de comercio electrónico, como SwiftGadget, enfrenta múltiples desafíos que pueden afectar su funcionalidad y éxito en el mercado. Uno de los principales problemas es la seguridad. Las aplicaciones PHP pueden ser vulnerables a ataques si no se implementan correctamente medidas de seguridad, lo que pone en riesgo la información de los usuarios y la integridad del sitio web.

Otro desafío significativo es laexperiencia de usuario. Una navegación intuitiva y tiempos de carga rápidos son esenciales para retener a los clientes y fomentar las compras.

Un error muy grave es no tener en cuenta la experiencia de usuario cuando se diseña una web. Preocúpate porque tu web sea fácil de usar, haz una prueba heurística de usabilidad o si tienes presupuesto haz una prueba de usuarios para comprobar que no existen errores graves en tu web que puedan afectar a tus visitantes. (e-intelligent, 2018)

Según, (Statista Research Department, 2024) “el creciente uso de internet y de las conexiones móviles ha propiciado un auténtico auge del tráfico digital concentrado por estas herramientas digitales. Más en concreto, Colombia, con más de 38 millones de usuarios, es uno de los mayores ejemplos del crecimiento experimentado en este sector”. Esto nos quiere decir que la compatibilidad móvil es crucial, especialmente en mercados como Colombia, donde la mayoría de los usuarios de internet accede a través de dispositivos móviles. Si un sitio web no está optimizado para dispositivos móviles, se arriesga a perder una parte significativa de su audiencia potencial.

Por último, la gestión de inventario en tiempo real es fundamental para evitar la venta de productos agotados y garantizar la satisfacción del cliente. La falta de actualización en el inventario puede llevar a experiencias negativas y afectar la reputación de la tienda en línea. Abordar estos problemas es esencial para SwiftGadget, en el competitivo mercado de la venta de dispositivos tecnológicos.

# Pregunta De Investigación

¿Cómo puede optimizarse la seguridad y la experiencia del usuario en una plataforma de comercio electrónico desarrollada en PHP?

# Objetivos

## Objetivo general

Analizar y desarrollar estrategias para optimizar la seguridad, la experiencia del usuario y la gestión del inventario en Swift Gadget, plataforma de comercio electrónico de venta de dispositivos tecnológicos.

## Objetivos específicos

Identificar los principales problemas de seguridad en plataformas e-commerce desarrolladas en PHP y proponer soluciones efectivas para proteger los datos de los clientes.

Evaluar la importancia de la experiencia del usuario en la retención de clientes y diseñar mejoras en la interfaz y funcionalidad de Swift Gadget.

Desarrollar un modelo eficiente de gestión de inventario en tiempo real para garantizar la disponibilidad de productos y mejorar la operatividad del sitio web.

Desarrollar e integrar un agente virtual de inteligencia artificial en Swift Gadget para mejorar la experiencia del usuario y optimizar la atención al cliente

# Introducción

El comercio electrónico en Colombia ha experimentado un crecimiento notable en los últimos años. Según (Mantilla, 2024) “Al cierre del 2023, el comercio electrónico en Colombia creció un 12.58%, según dio a conocer la Cámara Colombiana de Comercio Electrónico (CCCE). Ello representa un volumen de ventas de más de $62,5 billones en ventas online y $370,5 millones de transacciones digitales.” Este incremento refleja una adopción significativa de las compras digitales por parte de los consumidores colombianos.

En este contexto, Swift Gadget surge como una plataforma de comercio electrónico especializada en la venta de dispositivos tecnológicos, incluyendo periféricos, sillas gamer y gabinetes. Todo esto desarrollado en PHP, esta plataforma busca ofrecer una experiencia de usuario óptima y segura.

Sin embargo, el desarrollo de plataformas e-commerce en PHP presenta desafíos específicos. Uno de los principales es garantizar la seguridad del sitio, ya que las amenazas cibernéticas son una preocupación constante para las tiendas en línea. Entre ellos, encontramos vulnerabilidades o ataques, tales como, como inyecciones SQL o denegaciones de servicios (DDos). Que son muy frecuentas en plataformas e-commerce. Según (Zúñiga, 2024) “Los agujeros de seguridad que más mala fama le dan a PHP se podrían prevenir si somos precavidos con las conocidas como Inyecciones SQL. Podríamos evitarlo de una manera sencilla escapando de cualquier variable que podamos utilizar contra la base de datos. “Esto nos quiere decir que es fundamental implementar medidas de protección para evitar vulnerabilidades en el sistema y garantizar la privacidad de los usuarios. Ya que, un fallo en la seguridad podría exponer datos sensibles, afectando la confianza de los clientes y la reputación del sitio web.

PHP sigue en constante evolución y cada versión que sale optimiza y mejora su rendimiento, ofreciendo una mayor seguridad y compatibilidad con otros sistemas.

Seguridad. Seguir con versiones antiguas, como PHP 5.4 o PHP 5.5, hará que tengamos ataques de seguridad y que solo corregiremos si tenemos una versión actualizada. Atrás quedaron sus obsoletos parches que caducaron en 2016.

Rendimiento. Cuanto más rápido cargue una web, mejor clasificación obtendremos en los motores de búsqueda y mejores posiciones tendremos en las SERP (páginas de resultados). Las versiones más recientes de PHP son perfectas para incrementar esa velocidad, ya que son capaces de manejar más solicitudes por segundo. (Zúñiga, 2024)

En primer lugar, la seguridad en plataformas de comercio electrónico es fundamental. Muchas de las versiones de antiguas de PHP podrían no tener un gran soporte o escalabilidad. Por ende, lo mejor podría ser adaptar nuevas actualizaciones y metodologías, no solo con el fin de seguridad, si no también con la idea de destacar. Aplicando nuevas tecnologías, de esta manera destacando entre otros tipos de plataformas de comercio electrónico.

Además, se busca implementar un agente virtual de inteligencia artificial con el objetivo de mejorar la experiencia del usuario y optimizar la atención al cliente en Swift Gadget. Esta tecnología permitirá ofrecer respuestas rápidas y precisas a las consultas de los clientes, facilitando la navegación en la plataforma y mejorando la satisfacción del usuario.

Ilustración 1

Texto

El contenido generado por IA puede ser incorrecto.

Ilustración 2

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

# Análisis de Requerimientos

## Objetivo del Proyecto

Desarrollar e implementar mejoras en la plataforma e-commerce Swift Gadget, incluyendo optimización de seguridad, experiencia del usuario y gestión del inventario, además de la integración de un agente virtual de inteligencia artificial para mejorar la atención al cliente y la interacción con los usuarios.

## Requisitos Funcionales

* Gestión de productos: Este módulo permitirá a los usuarios agregar, modificar y eliminar productos en la tienda. Asimismo, mostrar detalles de cada producto (nombre, precio, descripción, imágenes, stock).
* Carrito de compras: Permitirá a los usuarios agregar y eliminar productos del carrito. Por otro lado, calcular el total de la compra en tiempo real.
* Sistema de autenticación (Opcional): Permitirá el registro e inicio de sesión de usuarios para recibir información sobre descuentos y stock.
* Métodos de pago: Integrar métodos de pago como tarjetas de crédito/débito y transferencias. Asimismo, confirmar pagos y generar comprobantes de este.
* Sistema de búsqueda y filtrado: Este módulo permitirá a los usuarios buscar productos por nombre o categoría.
* Gestión de inventario: Se actualizará automáticamente el stock después de cada compra.
* Agente virtual de IA: Responderá preguntas frecuentes sobre productos, pedidos y asistirá a los clientes en la navegación por el sitio web.

## Requisitos no Funcionales

* Seguridad: Para garantizar la seguridad de los datos, se podría implementar un cifrado de contraseñas y protección contra inyecciones SQL. Asimismo, se podría usar mejor un HTTPS para proteger la comunicación de datos.
* Escalabilidad y rendimiento: Asegúrese de que sea posible actualizar las características del proyecto sin problemas, sin impacto, y garantice el asesoramiento oportuno de expertos.
* Disponibilidad: Para brindar un buen servicio, se garantizaría que la plataforma esté en funcionamiento 24/7. Por otro lado, se implementaría copias de seguridad periódicas para evitar perdida de datos. Por medio de Instantáneas o Transacciones SQL.

## Diseño

### Arquitectura del Sistema

* El Sistema se basará en la Arquitectura de Cliente-Servidor
* Diseño de la Interfaz del Usuario (UI/UX)
* Experiencia de Usuario (UX)
* Seguridad en el Diseño

## Cliente-Servidor

* Cliente (Front-end): Desarrollado con HTML, CSS y JS, para la experiencia de usuario.
* Servidor (Back-end): Desarrollado con PHP, con el fin de manejar la lógica de negocio.
* Base de Datos: Se usará MySQL o PostgreSQL, con el fin de, almacenar información de productos, usuarios y pedidos. Asimismo, se implementará transacciones (BEGIN TRANSACTION, COMMIT, ROLLBACK) para evitar pérdida de datos en caso de errores.
* Servidor web: Apache.

## Diseño de Interfaz de Usuario (UI/UX)

* Wireframes.
* Página Principal.
* Página de Productos
* Carrito de Compras.
* Secciones de Pago.
* ChatBot (AgenteIA).
* Panel de Administración.

## Experiencia de Usuario (UX)

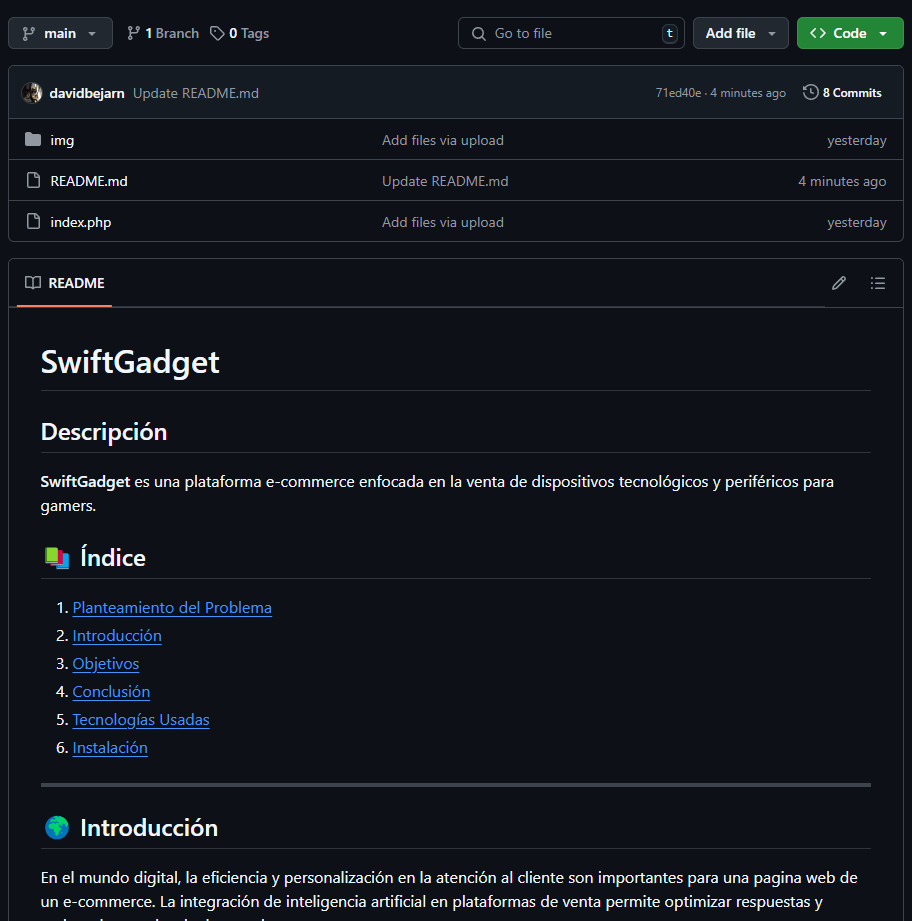
### Herramientas y Tecnologías

* Lenguajes (PHP, HTML, CSS, JS)
* Editores de texto (Visual Studio Code, Markdown, JupiterNoteBooks)
* Lenguajes de Bases de Datos (SQL)
* Despliegues (Docker, Vercel)

### Estrategia de Desarrollo

* Metodología: (SCRUM)
* Organización de código y repositorios: (Github)

Ilustración 3



*Nota. Repositorio público del proyecto (*[*https://github.com/davidbejarn/SwiftGadget*](https://github.com/davidbejarn/SwiftGadget)*)*

## Seguridad en el Diseño

* Uso de HTTPS y cifrado en la base de datos.
* Protección contra SQL Injection y Cross-Site Scripting (XSS).
* Implementación de autenticación de usuarios segura.

# 

# Proceso de Desarrollo

### Planificación:

* Reuniones periódicas para discutir el progreso, identificar obstáculos y ajustar las tareas.

### Desarrollo Ágil:

* Se utilizarán métodos flexibles, para garantizar una iteración rápida y flexibilidad en el proceso de desarrollo.
* Se realizarán revisiones de código y sesiones de programación en pares para fomentar la colaboración y el aprendizaje.

### Pruebas:

* Se realizarán pruebas e integración del sistema para garantizar la aplicación. Todos los componentes funcionan bien.

### Documentación:

* Crear guías de usuario e instrucciones para facilitar la implementación de aplicaciones.

### Revisar y Adaptar:

* Después de cada iteración, se revisará lo que se ha desarrollado y se recopilará comentarios de los usuarios o partes interesadas para realizar ajustes en futuras iteraciones.

## Prueba

### Tipo de Prueba

* Prueba unitaria.
* Pruebas de integración.
* Pruebas de sistema y aceptación.

## Implementación

### Preparación de la Implementación:

* Configuración del entorno (Servidor, Base de Datos).

### ***Estrategia de Lanzamiento***:

* Según calendario se desarrolla una estrategia de migración de datos si es necesario. (Esto está implícito durante el desarrollo de la aplicación).

### Monitoreo Posterior a la Implementación:

Se harán monitoreos para solución de problemas de implementación inicial de la aplicación.

## Mantenimiento

### Programa de Mantenimiento:

* Actualizaciones y parches de seguridad: Las actualizaciones se intentarán en horarios normales. En caso de que se requiera la instalación de una actualización o parche para resolver un problema crítico, se aplicará de inmediato.
* Administración de errores y fallas: Para la gestión de errores y fallas del servicio, se implementará un sistema de tickets de soporte, permitiendo un seguimiento adecuado y una resolución eficiente de los incidentes reportados.

## Documentación continua:

* La documentación técnica y de usuario, estará actualizada (cualquier cambio se reflejará en el manual del usuario y se proporciona capacitación, dicho manual y documentación se encontrará en repositorios almacenados en GitHub).

## Atención al cliente:

* Incorporación de un agente virtual basado en IA, se tiene como objetivo optimizar el soporte al cliente, ofreciendo asistencia en tiempo real sobre disponibilidad de productos, tiempos de entrega, resolución de problemas técnicos y guía por la página web.

# Referencias

e-intelligent. (26 de Abril de 2018). *e-intelligent*. Obtenido de e-intelligent: https://www.e-intelligent.es/es/blog/consultoria-web-y-marketing-online/los-10-errores-mas-comunes-en-paginas-web-de-empresas-y-pymes/

Mantilla, R. (20 de Junio de 2024). *enviame*. Obtenido de enviame: https://enviame.io/ecommerce-colombia/

Statista Research Department. (10 de Septiembre de 2024). *statista*. Obtenido de statista: https://es.statista.com/temas/10524/redes-sociales-en-colombia/

Zúñiga, F. G. (18 de Octubre de 2024). *arsys*. Obtenido de arsys: https://www.arsys.es/blog/tips-seguridad-php

Gómez, P., Torres, L., & Rodríguez, M. (2020). *Estrategias de atención al cliente en el comercio electrónico: Impacto en la fidelización de consumidores.* Editorial Universitaria.

Pérez, J., & Ramírez, C. (2021). *Chatbots e inteligencia artificial en la atención al cliente: Un enfoque práctico.* TechPress.

Martínez, R. (2019). *Tendencias en servicio al cliente: Automatización y personalización en la era digital.* Revista de Negocios y Tecnología, 15(2), 45-62.

López, A., & Fernández, S. (2022). *La omnicanalidad en la experiencia del cliente: Estrategias para mejorar la comunicación empresarial.* Global Marketing Journal, 18(3), 78-95.